AMENDMENTS TO THE CLAIMS

Listing of Claims:

1. (Previously presented) A method comprising:

first causing, by a first module, a second module to be logically chained to the first module to form or expand a thread comprised of modules, the modules of the thread including at least the first and second modules, enabling the first and second modules to be executed as part of the thread and to share an execution context of the thread;

second causing, by the first module, the second module to register first events of interest to the second module with an event notification service;

waiting, by the first module, for notification of occurrence of one or more of the first events; and

third causing, by the first module, the second module to process an occurred one of the first events.

2. (Previously presented) The method of claim 1, wherein the first causing further includes invoking a thread module chaining service, the thread module chaining service having a control data structure of the thread to logically associate the second module with the first module, the control data structure including a first pointer to a first set of executable instructions of the second module designed to register the first events with the event notification service.

3. (Previously presented) The method of claim 2, wherein the second causing further includes invoking the thread module chaining service to orchestrate registration of events of interest to logically chained modules of the thread with the event notification service, the thread module chaining service retrieving the first pointer to the first set of executable instructions, and causing the first set of executable instructions to be executed, using the first pointer to locate the first set of executable instructions.

4.   (Previously presented) The method of claim 2, wherein the control data structure further includes a second pointer to a second set of executable instructions designed to process the occurred one of the first events.

5.   (Previously presented) The method of claim 4, wherein the third causing further comprises invoking the thread module chaining service to orchestrate processing of the occurred event by the logically chained modules of the thread, the thread module chaining service retrieving the second pointer to the second set of executable instructions, and causing the second set of executable instructions to be executed, using the second pointer to locate the second set of executable instructions.

6.   (Previously presented) The method of claim 2, wherein the method further comprises
        detecting, by the first module, for a thread termination condition; and
        fourth causing, by the first module the thread to be terminated after detecting the thread termination condition.

7.   (Previously presented) The method of claim 6, wherein the fourth causing further comprises invoking the thread module chaining service to orchestrate thread termination clean up by the logically chained modules of the thread, the thread module chaining service retrieving the a third pointer, stored in the control data structure, to a third set of executable instructions of the second module, the third set of executable instructions designed to perform termination clean up for the second module, and causing the third set of executable instructions to be executed, using the third pointer to locate the third set of executable instructions.

8.   (Previously presented) The method of claim 1, further comprising
        fourth causing, by the first module, a third module to be logically chained to the first and second modules to expand the thread, enabling the first, second, and third modules to be executed as part of the thread and to share an execution context of the thread.

9.   (Previously presented) A method comprising:

first causing, by a first module, a second module to be logically chained to the first module to form or expand a thread comprised of modules, the modules of the thread including at least the first and second modules, enabling the first and second modules to be executed as part of the thread and to share an execution context of the thread;

waiting, by the first module, for notification of occurrence of one or more of first events of interest to the second module;

second causing, by the first module, the second module to process an occurred one of the first events;

detecting, by the first module, for a thread termination condition; and

third causing, by the first module, the thread to be terminated after detecting the thread termination condition.

10.   (Previously presented) The method of claim 9, wherein said second causing further comprises invoking a thread module chaining service to orchestrate processing of the occurred event by logically chained modules of the thread, the thread module chaining service retrieving a first pointer to a first set of executable instructions of the second module from a control data structure, the first set of executable instructions designed to process the occurred one of the first events, and causing the first set of executable instructions to be executed, using the first pointer to locate the first set of executable instructions.

11.   (Cancelled)

12.   (Previously presented) The method of claim 9, wherein the third causing further comprises invoking a thread module chaining service to orchestrate thread termination clean up by logically chained modules of the thread, the thread module chaining service retrieving a first pointer to a first set of executable instructions of the second module from a control data structure, the first set of executable instructions designed to perform termination clean up for the second module, and causing the first set of executable instructions to be executed, using the first pointer to locate the second set of executable instructions.

13. (Previously presented) The method of claim 9, further comprising fourth causing, by the first module, a third module to be logically chained to the first and second modules to expand the thread, enabling the first, second, and third modules to be executed as part of the thread and to share an execution context of the thread.

14-18. (Cancelled)

19. (Previously presented) A computing device comprising:

a processor;

a first module to be operated by the processor to cause a second module to be logically chained to the first module to form or expand a thread comprised of modules, the modules of the thread including at least the first and second modules, enabling the first and second modules to be executed as part of the thread and to share an execution context of the thread; and

the second module to be operated by the processor, the second module having at least a selected one of a first set of executable instructions to register events of interest to the second module, a second set of executable instructions to process an occurred one of the events of interest, and a third set of executable instructions to perform clean up during thread termination.

20. (Previously presented) The computing device of claim 19, wherein the second module further includes at least a corresponding one of a first pointer, a second pointer, and a third pointer pointing to the first set, the second set, and the third set of executable instructions respectively.

21. (Original) The computing device of claim 19, wherein the computing device comprises a UPNP control point.

22-24. (Cancelled)

25-31. (Cancelled)